It is claimed:

1. A computer-implemented method that authors information stored in a document, wherein a client computer processes the information at run-time, comprising the steps of:

5   receiving information about a first control that has both property types and property values that define how the first control appears in the document,

wherein the first control generates both design-time information and run-time information, wherein the run-time information is processed at run-time, wherein the design-time information is not used during run-time;

10   creating a second control whose property types substantially match the property types of the first control;

generating an interface for modifying the property values of the second control; and

using the second control's modified property values to update the property
15 values of the first control in the document.

2. The method of claim 1 wherein the design-time information is used during authoring of the information.

20   3. The method of claim 1 wherein the first and second controls are Design-Time Controls.

4. The method of claim 3 wherein a software application authors the information and whose extensibility interfaces do not natively support a Design-Time Control.

5. The method of claim 4 further comprising the steps of:

5   providing a data communication connection to the software application through at least one of the extensibility interfaces;

receiving information about the first control through the data communication connection; and

sending the second control's modified property values to the software

10   application through the data communication connection in order to update the property values of the first control in the document.

6. The method of claim 1 wherein a graphic image represents the first control in the document.

15

7. The method of claim 6 further comprising the step of:

receiving information about the first control after the graphic image of the first control has been activated by a user in order to edit the property values of the first control.

20

8. The method of claim 1 further comprising the step of:

receiving an identifier to indicate type of Design Time Control to create as

the second control, wherein the identifier is received when the property values of the first

control are empty.

5   9. The method of claim 8 wherein the identifier is retrieved from an operating system

registry.

10. The method of claim 1 further comprising the steps of:

receiving the property values of the first control, wherein at least one of

10   the property values is not empty; and

determining type of Design Time Control to create as the second control

based upon the property values.

11. The method of claim 1 further comprising the steps of:

15   receiving at least a portion of the run-time information of the first control;

parsing the run-time information in order to determine the property values

of the first control; and

determining type of Design-Time Control to create as the second control

based upon the determined property values.

20

12. The method of claim 11 wherein the run-time information includes Hypertext

Markup Language (HTML) run-time instructions.

24

13. The method of claim 1 wherein the run-time information includes Hypertext Markup Language (HTML) run-time instructions, said method further comprising the steps of:

receiving the HTML run-time instructions of the first control;

parsing the HTML run-time instructions to obtain a properties map,

metadata tag text, a control class identifier, a program identifier, and inner HTML information; and

determining type of Design-Time Control to create as the second control based upon the parsed HTML run-time instructions.

14. The method of claim 1 further comprising the step of:

creating a hidden window to store the second control.

15. The method of claim 1 further comprising the step of:

generating a computer-human interface for modifying the property values of the second control.

16. The method of claim 15 wherein the run-time information includes Hypertext Markup Language (HTML) run-time instructions, said method further comprising the steps of:

receiving the HTML run-time instructions of the first control;

parsing the HTML run-time instructions to obtain a properties map, metadata tag text, a control class identifier, a program identifier, and inner HTML information;

25

determining type of Design-Time Control to create as the second control

based upon the parsed HTML run-time instructions; and

determining what properties to display in the computer-human interface

based upon the parsed HTML run-time instructions.

5

17. The method of claim 1 further comprising the step of:

converting the second control's modified property values into run-time

information to be used within the document.

10   18. The method of claim 1 further comprising the step of:

converting the second control's modified property values into Hypertext

Markup Language (HTML) run-time instructions to be used within the document.

19. The method of claim 1 wherein the client computer receives the document over a

15   network in order to processes the information.

20. The method of claim 19 wherein the network is selected from the group consisting of

local area networks, wide area networks, global networks, and combinations thereof.

20   21. The method of claim 1 wherein the document is an Hypertext Markup Language

(HTML) document.

22. A computer-implemented apparatus that authors information stored in a document, wherein a client computer processes the information at run-time, comprising:

a Design-Time Control wrapper module that receives information about a

5 first control, wherein the first control has both property types and property values that define how the first control appears in the document,

wherein the first control generates both design-time information and run-time information, wherein the run-time information is processed at run-time, wherein the design-time information is not used during run-time;

10 a window connected to the Design-Time Control wrapper within which a second control is created, wherein the second control's property types substantially match the property types of the first control; and

a properties interface connected to the window to modify the property values of the second control,

15 whereby the second control's modified property values are used to update the property values of the first control in the document.

23. The apparatus of claim 22 wherein the design-time information is used during authoring of the information.

24. The apparatus of claim 22 wherein the first and second controls are Design-Time
5   Controls.

25. The apparatus of claim 24 wherein a software application authors the information and whose extensibility interfaces do not natively support a Design-Time Control.

10   26. The apparatus of claim 25 further comprising:

        a run-time translator module connected to the Design-Time Control wrapper module, wherein the run-time translator includes a data communication connection to the software application through at least one of the extensibility interfaces,

        wherein the run-time translator module receives information about the first
15   control in order to provide the information to the Design-Time Control wrapper module.

27. The apparatus of claim 22 wherein a graphic image represents the first control in the document.

20   28. The apparatus of claim 22 wherein the Design-Time Control wrapper module receives an identifier to indicate type of Design-Time Control to create as the second control, wherein the identifier is received when the property values of the first control are empty.

29. The apparatus of claim 28 wherein the identifier is from an operating system registry.

30. The apparatus of claim 22 wherein the Design-Time Control wrapper module

5   receives property values of the first control, wherein at least one of the property values is

not empty, wherein the Design-Time Control wrapper module determines type of Design-

Time Control to create as the second control based upon the received property values.

31. The apparatus of claim 22 wherein the Design-Time Control wrapper module parses

10   the run-time information of the first control in order to determine the property values of

the first control, said apparatus further comprising:

        a properties data structure to store the property values,

        wherein type of Design-Time Control to create as the second control is

based upon the stored property values.

15

32. The apparatus of claim 31 wherein the run-time information includes Hypertext

Markup Language (HTML) run-time instructions.

33. The apparatus of claim 32 further comprising:

20   a properties map data structure connected to the Design-Time Control

wrapper module to store names and values of the properties parsed from the HTML run-

time instructions,

wherein the interface includes the property names and values stored in the properties map data structure.

34. The apparatus of claim 32 further comprising:

   5   a metadata tag text data structure connected to the Design-Time Control wrapper module to store a header portion contained in the HTML run-time instructions.

35. The apparatus of claim 32 further comprising:

   a control class identification data structure connected to the Design-Time

   10 Control wrapper module to store a control classification identifier contained in the HTML run-time instructions.

36. The apparatus of claim 32 further comprising:

   a program identifier data structure connected to the Design-Time Control

   15 wrapper module to store a program identifier contained in the HTML run-time instructions,

   wherein the stored program identifier is used to determine what type of Design-Time Control to create within the window.

   20 37. The apparatus of claim 32 further comprising:

   an inner HTML data structure connected to the Design-Time Control wrapper module to store substantially a complete HTML text representation of the first control.

30

38. The apparatus of claim 22 wherein the window is a hidden window to store the second control.

5   39. The apparatus of claim 22 wherein the properties interface is a computer-human interface that modifies the property values of the second control.

40. The apparatus of claim 39 wherein the run-time information includes Hypertext Markup Language (HTML) run-time instructions, said apparatus further comprising:

10   means for receiving the HTML run-time instructions of the first control;

means for parsing the HTML run-time instructions to obtain a properties map, metadata tag text, a control class identifier, a program identifier, and inner HTML information;

means for determining type of Design-Time Control to create as the

15   second control based upon the parsed HTML run-time instructions; and

means for determining what properties to display in the computer-human interface based upon the parsed HTML run-time instructions.

41. The apparatus of claim 22 further comprising:

20   means for converting the second control's modified property values into run-time information to be used within the document.

42. The apparatus of claim 22 further comprising:

means for converting the second control's modified property values into Hypertext Markup Language (HTML) run-time instructions to be used within the document.

5   43. The apparatus of claim 22 wherein the client computer receives the document over a network in order to processes the information.

44. The apparatus of claim 43 wherein the network is selected from the group consisting of local area networks, wide area networks, global networks, and combinations thereof.

10

45. The apparatus of claim 22 wherein the document is an Hypertext Markup Language (HTML) document.

46. The apparatus of claim 22 wherein the Design-Time Control wrapper module

15   handles a plurality of different types of Design-Time Controls that are used within the document.